

Sửa lỗi trong Git

Đã là con người thì ai cũng phạm sai lầm. Vì thế mỗi VCS đều cung cấp một tính năng để sửa lỗi tại một điểm nào đó. Git cung cấp một tính năng mà chúng ta có thể sử dụng để undo các chỉnh sửa mà đã làm trên repository nội bộ.

Giả sử người sử dụng tình cờ thực hiện một vài thay đổi tới repository nội bộ và sau đó muốn undo những thay đổi này. Trong những trường hợp như vậy, hoạt động revert đóng một vai trò quan trọng.

Trả lại những thay đổi chưa được commit

Giả sử Jerry tình cờ chỉnh sửa một file từ trong repository nội bộ của anh ta. Nhưng anh ta muốn undo lại tùy chỉnh này. Để giải quyết tình huống này, chúng ta sử dụng lệnh git checkout. Chúng ta có thể sử dụng lệnh này để trả lại những nội dung của một file.

```
[jerry@CentOS src]$ pwd
/home/jerry/jerry_repo/project/src

[jerry@CentOS src]$ git status -s
M string_operations.c

[jerry@CentOS src]$ git checkout string_operations.c

[jerry@CentOS src]$ git status -s
```

Xa hơn nữa, chúng ta có thể sử dụng lệnh git checkout để đạt được các file bị xóa từ repository nội bộ. Giả sử Tom xóa một file từ repository nội bộ và anh ta muốn sử dụng file này sau đó. Chúng ta có thể thực hiện được điều này bằng cách sử dụng lệnh giống trên.

```
[tom@CentOS src]$ pwd
/home/tom/top_repo/project/src

[tom@CentOS src]$ ls -l
Makefile
string_operations.c

[tom@CentOS src]$ rm string_operations.c
```

```
[tom@CentOS src]$ ls -l
Makefile

[tom@CentOS src]$ git status -s
D string_operations.c
```

Git đang chỉ ký tự D trước tên file. Điều này chỉ rằng file đã được xóa từ repository nội bộ.

```
[tom@CentOS src]$ git checkout string_operations.c

[tom@CentOS src]$ ls -l
Makefile
string_operations.c

[tom@CentOS src]$ git status -s
```

Ghi chú: Chúng ta có thể thực hiện tất cả những hoạt động này trước hoạt động commit.

Dỡ bỏ những thay đổi từ khu vực tổ chức

Chúng ta đã nhìn thấy khi chúng ta chúng ta thực hiện một hoạt động thêm, các file di chuyển từ repository nội bộ tới khu vực tổ chức. Nếu một người sử dụng tình cờ chỉnh sửa một file và thêm nó vào trong khu vực tổ chức, anh ta có thể trả lại những thay đổi, bằng cách sử dụng lệnh git checkout.

Trong Git, có một điểm con trỏ HEAD mà luôn luôn trỏ tại commit mới nhất. Nếu bạn muốn undo một sự thay đổi từ khu vực tổ chức, thì khi đó bạn có thể sử dụng lệnh git checkout, nhưng với lệnh này, bạn phải cung cấp thêm một tham số, i.e., điểm trỏ HEAD. tham số con trỏ commit thêm vào này chỉ thị lệnh git checkout để reset cây làm việc và cũng để dỡ bỏ những thay đổi được tổ chức.

Giả sử Tom chỉnh sửa một file từ repository nội bộ. Nếu chúng ta quan sát trạng thái của file này, nó sẽ chỉ rằng file này được chỉnh sửa nhưng không thêm vào trong khu vực tổ chức.

```
tom@CentOS src]$ pwd
/home/tom/top_repo/project/src
# Unmodified file

[tom@CentOS src]$ git status -s
```

```
# Modify file and view it's status.  
[tom@CentOS src]$ git status -s  
M string_operations.c  
  
[tom@CentOS src]$ git add string_operations.c
```

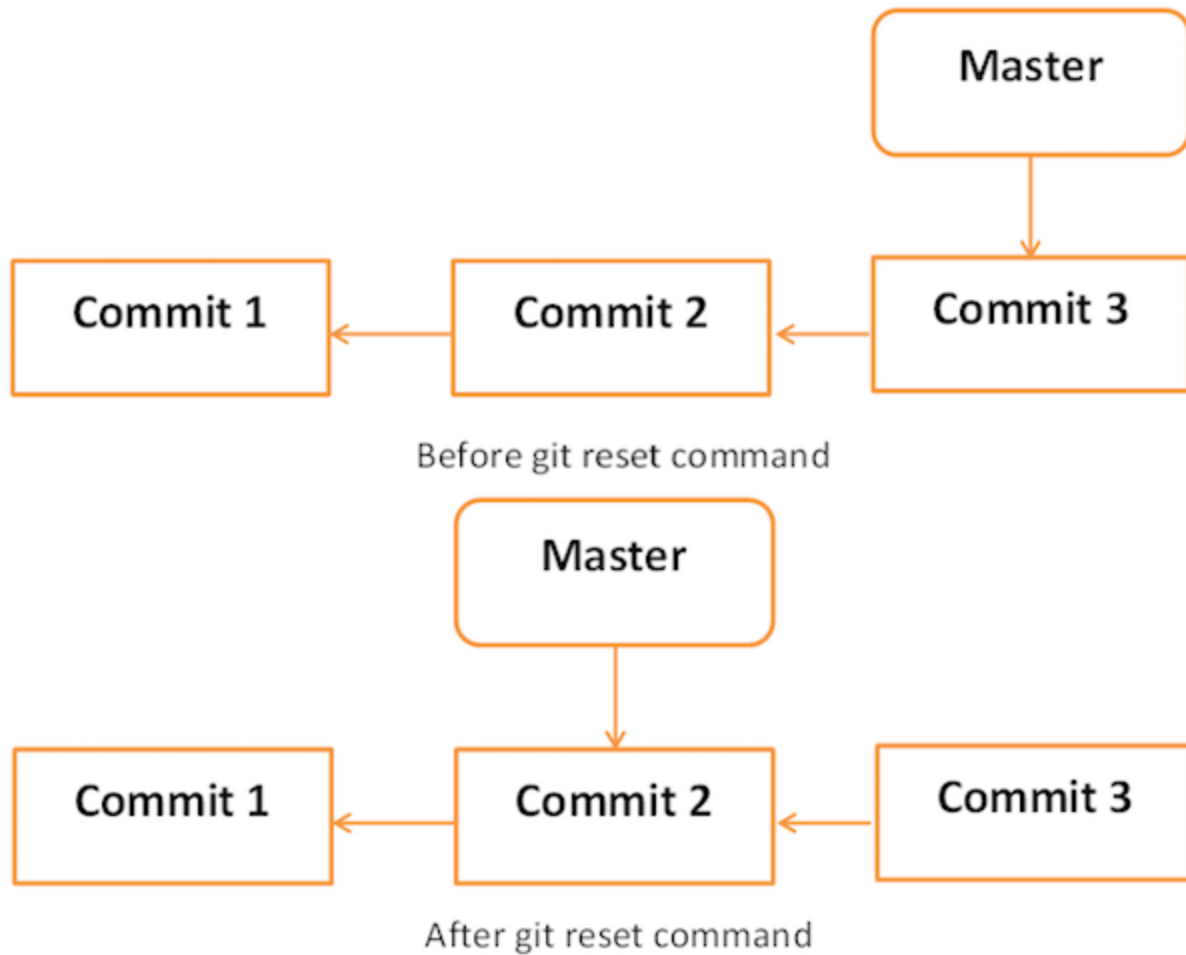
Git status chỉ rằng file hiện diện trong khu vực tổ chức, bây giờ trả lại nó bằng cách sử dụng lệnh git command và quan sát trạng thái của file được trả lại.

```
[tom@CentOS src]$ git checkout HEAD -- string_operations.c  
  
[tom@CentOS src]$ git status -s
```

Di chuyển điểm trở HEAD với git reset

Sau khi thực hiện một số thay đổi, bạn có thể quyết định dỡ bỏ những thay đổi này. Lệnh git reset được sử dụng để reset hoặc trả lại những thay đổi. Chúng ta có thể thực hiện ba kiểu khác nhau của hoạt động reset.

Sơ đồ dưới chỉ tiến trình của lệnh git reset.



Soft

Mỗi nhánh có một điểm trỏ HEAD, mà trỏ vào commit mới nhất. Nếu chúng ta sử dụng lệnh git reset với tùy chọn --soft theo sau bằng ID commit, thì khi đó nó sẽ chỉ reset điểm trỏ HEAD mà không phá hủy bất cứ thứ gì.

Tệp **.git/refs/heads/master** giữ ID commit của điểm trỏ HEAD. Chúng ta có thể kiểm tra nó bằng cách sử dụng lệnh git log-1.

```
[jerry@CentOS project]$ cat .git/refs/heads/master  
577647211ed44fe2ae479427a0668a4f12ed71a1
```

Bây giờ, quan sát các ID commit mới nhất, mà sẽ kết nối với ID commit trên.

```
[jerry@CentOS project]$ git log -2
```

Lệnh trên sẽ tạo ra kết quả sau:

```
commit 577647211ed44fe2ae479427a0668a4f12ed71a1
Author: Tom Cat <tom@tutorialspoint.com>
Date: Wed Sep 11 10:21:20 2013 +0530
```

Removed executable binary

```
commit 29af9d45947dc044e33d69b9141d8d2dad37cc62
Author: Jerry Mouse <jerry@tutorialspoint.com>
Date: Wed Sep 11 10:16:25 2013 +0530
```

Added compiled binary

Hãy cùng chúng tôi reset điểm trở HEAD.

```
[jerry@CentOS project]$ git reset --soft HEAD~
```

Bây giờ chúng ta vừa reset điểm trở HEAD trở lại sau một vị trí. Chúng ta kiểm tra lại nội dung của tệp **.git/refs/heads/master file**.

```
[jerry@CentOS project]$ cat .git/refs/heads/master
29af9d45947dc044e33d69b9141d8d2dad37cc62
```

ID commit từ file được thay đổi, bây giờ thăm tra nó bằng cách kiểm tra thông báo commit.

```
jerry@CentOS project]$ git log -2
```

Lệnh trên sẽ tạo ra kết quả sau:

```
commit 29af9d45947dc044e33d69b9141d8d2dad37cc62
Author: Jerry Mouse <jerry@tutorialspoint.com>
Date: Wed Sep 11 10:16:25 2013 +0530
```

Added compiled binary

```
commit 94f7b26005f856f1a1b733ad438e97a0cd509c1a
Author: Jerry Mouse <jerry@tutorialspoint.com>
Date: Wed Sep 11 10:08:01 2013 +0530
```

Added Makefile and renamed strings.c to string_operations.c

Tùy chọn mixed

Lệnh git reset với tùy chọn --mixed trả lại những thay đổi từ trong khu vực tổ chức mà chưa được commit. Nó chỉ trả lại những thay đổi từ khu vực tổ chức. Những thay đổi thực sự đối với bản sao làm việc của một file không bị ảnh hưởng. Lệnh git reset mặc định tương đương với git reset --mixed.

Tùy chọn hard

Nếu bạn sử dụng tùy chọn --hard với lệnh git reset, nó sẽ xóa khu vực tổ chức; nó sẽ reset điểm trở HEAD tới những commit mới nhất của ID commit cụ thể và xóa các thay đổi file nội bộ.

Chúng ta kiểm tra ID commit.

```
[jerry@CentOS src]$ pwd
/home/jerry/jerry_repo/project/src

[jerry@CentOS src]$ git log -1
```

Lệnh trên sẽ tạo kết quả sau:

```
commit 577647211ed44fe2ae479427a0668a4f12ed71a1
Author: Tom Cat <tom@tutorialspoint.com>
Date: Wed Sep 11 10:21:20 2013 +0530

Removed executable binary
```

Jerry chỉnh sửa một file bằng cách thêm một lời bình dòng đơn tại vị trí đầu của file.

```
[jerry@CentOS src]$ head -2 string_operations.c
/* This line be removed by git reset operation */
#include <stdio.h>
```

Anh ta kiểm tra nó bằng cách sử dụng lệnh git status.

```
[jerry@CentOS src]$ git status -s
M string_operations.c
```

Jerry thêm file được chỉnh sửa này tới khu vực tổ chức và kiểm tra nó với lệnh git status.

```
[jerry@CentOS src]$ git add string_operations.c
[jerry@CentOS src]$ git status
```

Lệnh trên sẽ tạo kết quả sau:

```
# On branch master
# Changes to be committed:
# (use "git reset HEAD <file>..." to unstage)
#
#
modified: string_operations.c
#
```

Git status đang chỉ rằng file đang hiện diện trong khu vực tổ chức. Bây giờ, reset điểm trở HEAD với tùy chọn --hard.

```
[jerry@CentOS src]$ git reset --hard 577647211ed44fe2ae479427a0668a4f12ed71a1

HEAD is now at 5776472 Removed executable binary
```

Lệnh git reset thực hiện thành công, mà sẽ trả lại file từ khu vực tổ chức cũng như dỡ bỏ bất kỳ những thay đổi cục bộ nào đã thực hiện với file.

```
[jerry@CentOS src]$ git status -s
```

Git status đang chỉ rằng file đã được dỡ bỏ khỏi khu vực tổ chức.

```
[jerry@CentOS src]$ head -2 string_operations.c
#include <stdio.h>
```

Lệnh head cũng chỉ rằng hoạt động reset đã dỡ bỏ những thay đổi cục bộ.